



**UNIVERSIDAD NACIONAL DE PIURA**  
**Facultad de Ingeniería Industrial**  
**Escuela Profesional de Ingeniería Informática**



## **INFORME DE INVESTIGACIÓN**

### **PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL**

**Presentada por:**

Bach. Chinguel Zurita Juliana De Lourdes

Bach. Gallardo Vasquez Gleydi Smith

Bach. Marchena Estrada Yessenia Edith

Bach. Silva Otero Teresa Del Carmen

### **PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO**

**Línea de investigación**  
**Informática, Electrónica y Telecomunicaciones**

**Sublínea de investigación**  
**Automatización y Control**

**Piura, Perú**

**2019**



**UNIVERSIDAD NACIONAL DE PIURA**  
**Facultad de Ingeniería Industrial**  
**Escuela Profesional de Ingeniería Informática**



## **INFORME DE INVESTIGACIÓN**

### **PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL**

**Línea de investigación**  
**Informática, Electrónica y Telecomunicaciones**

Elaborado por:

Bach. Chinguel Zurita Juliana De Lourdes

Bach. Gallardo Vasquez Gleydi Smith

Bach. Marchena Estrada Yessenia Edith

Bach. Silva Otero Teresa Del Carmen

Asesor:

Mg. Hebert Eduardo Espino Aguirre

## **DECLARACIÓN JURADA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN**

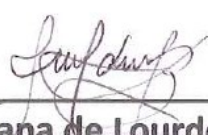
Yo, **CHINGUEL ZURITA JULIANA DE LOURDES** identificada con **DNI N° 72112920** y domiciliada en A.H. La Primavera - Etapa II Mz. Q Lt. 28, distrito de Castilla, provincia de Piura, Bachiller de la **Escuela Profesional de Ingeniería Informática** de la **Facultad de Ingeniería Industrial**:

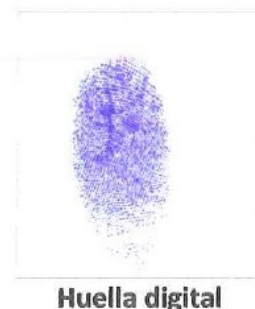
### **PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL**

**DECLARO BAJO JURAMENTO:** que la tesina que presento es original e inédita, no siendo copia parcial ni total de una tesis desarrollada y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporcione, me sujeto a los alcances de los establecido en el Art. N° 411 del Código Penal, concordante con el Art. 32° de la Ley N° 27444 y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo que digo, firmo la presente.

Piura, 19 de julio de 2019

  
**Bach. Juliana de Lourdes Chinguel Zurita**  
**DNI N°72112920**



**Huella digital**

**Artículo 411.-** El que, en un procedimiento administrativo, hace una falsa declaración en relación a hechos o circunstancias que le corresponde probar, violando la presunción de veracidad establecida por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 411, Inciso 4.12 del Reglamento del Registro Nacional de Trabajos de Investigación para Optar Grados Académicos y Títulos Profesionales – RENATI Resolución de Consejo Directivo N° 033-2016-SUNEDU/CD.

**DECLARACIÓN JURADA  
DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN**

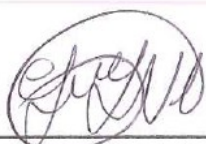
Yo, **GALLARDO VASQUEZ GLEYDI SMITH** identificada con **DNI N° 48268405** y domiciliada en A.H. Tacalá Mz. L Lt. 13, distrito de Castilla, provincia de Piura, Bachiller de la **Escuela Profesional de Ingeniería Informática de la Facultad de Ingeniería Industrial**:

**PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS  
DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL**

**DECLARO BAJO JURAMENTO:** que la tesina que presento es original e inédita, no siendo copia parcial ni total de una tesis desarrollada y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporciono, me sujeto a los alcances de los establecido en el Art. N° 411 del Código Penal, concordante con el Art. 32° de la Ley N° 27444 y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo que digo, firmo la presente.

Piura, 19 de julio de 2019



**Bach. Gleydi Smith Gallardo Vasquez**  
DNI N°48268405



**Huella digital**

**Artículo 411.-** El que, en un procedimiento administrativo, hace una falsa declaración en relación a hechos o circunstancias que le corresponde probar, violando la presunción de veracidad establecida por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 411, Inciso 4.12 del Reglamento del Registro Nacional de Trabajos de Investigación para Optar Grados Académicos y Títulos Profesionales – RENATI Resolución de Consejo Directivo N° 033-2016-SUNEDU/CD.



**DECLARACIÓN JURADA  
DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN**

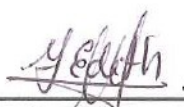
Yo, **MARCHENA ESTRADA YESSENIA EDITH** identificada con **DNI N° 47442252** y domiciliada en A.H. Ciudad del Sol Mz. M Lt. 6, distrito de Veintiséis de Octubre, provincia de Piura, Bachiller de la **Escuela Profesional de Ingeniería Informática de la Facultad de Ingeniería Industrial**:

**PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS  
DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL**

**DECLARO BAJO JURAMENTO:** que la tesina que presento es original e inédita, no siendo copia parcial ni total de una tesis desarrollada y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporciono, me sujeto a los alcances de lo establecido en el Art. N° 411 del Código Penal, concordante con el Art. 32° de la Ley N° 27444 y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo que digo, firmo la presente.

Piura, 19 de julio de 2019



**Bach. Yessenia Edith Marchena Estrada**

**DNI N°47442252**



**Huella digital**

**Artículo 411.-** El que, en un procedimiento administrativo, hace una falsa declaración en relación a hechos o circunstancias que le corresponde probar, violando la presunción de veracidad establecida por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 411, Inciso 4.12 del Reglamento del Registro Nacional de Trabajos de Investigación para Optar Grados Académicos y Títulos Profesionales – RENATI Resolución de Consejo Directivo N° 033-2016-SUNEDU/CD.

**DECLARACIÓN JURADA  
DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN**

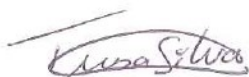
Yo, **SILVA OTERO TERESA DEL CARMEN** identificada con **DNI N° 47908660** y domiciliada en A.H. San Martín Av. María Galán Mz K15 Lt. 13, distrito de Veintiséis de Octubre, provincia de Piura, Bachiller de la **Escuela Profesional de Ingeniería Informática de la Facultad de Ingeniería Industrial**:

**PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS  
DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL**

**DECLARO BAJO JURAMENTO:** que la tesina que presento es original e inédita, no siendo copia parcial ni total de una tesis desarrollada y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporciono, me sujeto a los alcances de los establecido en el Art. N° 411 del Código Penal, concordante con el Art. 32° de la Ley N° 27444 y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo que digo, firmo la presente.

Piura, 19 de julio de 2019



---

**Bach. Teresa del Carmen Silva Otero**

**DNI N°47908660**



**Huella digital**

**Artículo 411.-** El que, en un procedimiento administrativo, hace una falsa declaración en relación a hechos o circunstancias que le corresponde probar, violando la presunción de veracidad establecida por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 411, Inciso 4.12 del Reglamento del Registro Nacional de Trabajos de Investigación para Optar Grados Académicos y Títulos Profesionales – RENATI Resolución de Consejo Directivo N° 033-2016-SUNEDU/CD.





UNIVERSIDAD NACIONAL DE PIURA  
PROGRAMA DE ACTUALIZACIÓN PROFESIONAL  
PATPRO EN INGENIERÍA INFORMÁTICA  
VERSIÓN XIX - 2019



**ACTA DE EVALUACIÓN DEL INFORME DE  
INVESTIGACIÓN**

Los Miembros del Jurado Calificador del Informe de Investigación denominado **"PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE COMPUTO A TRAVÉS DE UN DISPOSITIVO MOVIL"**, presentado por los bachilleres: **GALLARDO VASQUEZ GLEYDI SMITH, MARCHENA ESTRADA YESSSENIA EDITH, SILVA OTERO TERESA DEL CARMEN y CHINGUEL ZURITA JULIANA DE LOURDES**, participantes del **Programa de actualización para Titulación Profesional en la Especialidad de Ingeniería Informática Versión XIX 2019**; asesorados por el Mg. Hebert Espino Aguirre. Revisado y absueltas las observaciones formuladas por el Jurado Calificador, lo declaran:

*APROBADO*

Con la nota:

<b>GALLARDO VASQUEZ GLEYDI SMITH</b>	<u>13</u>
<b>MARCHENA ESTRADA YESSSENIA EDITH</b>	<u>13</u>
<b>SILVA OTERO TERESA DEL CARMEN</b>	<u>13</u>
<b>CHINGUEL ZURITA JULIANA DE LOURDES</b>	<u>13</u>



Piura, 13 de julio del 2019

*[Signature]*  
Mg. DUBERT REYES VASQUEZ  
Miembro del Jurado

*[Signature]*  
Dr. RIGO REQUENA FLORES  
Miembro del Jurado

*[Signature]*  
Mg. JUAN MANUEL JACINTO SANDOVAL  
Miembro del Jurado



**UNIVERSIDAD NACIONAL DE PIURA**  
**Facultad de Ingeniería Industrial**  
**Escuela Profesional de Ingeniería Informática**



## **INFORME DE INVESTIGACIÓN**

### **PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL**

**Línea de Investigación**  
**Informática, Electrónica y Telecomunicaciones**

**Aprobado por los jurados:**

---

**MG. DUBER REYES VÁSQUEZ**  
Miembro del Jurado

---

**DR. RIGO FÉLIX REQUENA FLORES**  
Miembro del Jurado

---

**MG. JUAN MANUEL JACINTO SANDOVAL**  
Miembro del Jurado



## **DEDICATORIA**

Este trabajo lo dedicamos a Dios por darnos la fortaleza y crecimiento intelectual; a nuestros queridos padres por su apoyo incondicional y por siempre confiar en nosotras en que con esfuerzo y dedicación podemos lograr nuestras metas.

## **AGRADECIMIENTO**

Queremos expresar nuestro agradecimiento al Ing. Hebert Eduardo Espino Aguirre, por asesorarnos en este proyecto de investigación y a todas las personas que contribuyeron con sus aportes y sugerencias constructivas para poder culminar con éxito este proceso.

## RESUMEN

El desarrollo del presente trabajo de investigación está basado en aplicar el Internet de las Cosas (IoT) para el diseño de un prototipo de un sistema de control de iluminación para centros de cómputo a través de una aplicación desarrollada en un sistema operativo Androide, que está instalada en un teléfono móvil, el cual se intercomunica por medio de un módulo Bluetooth, haciendo uso de un sensor detector de movimiento que envía una señal de alerta a la aplicación móvil, con el objetivo de controlar el suministro de energía eléctrica para la iluminación de un centro de cómputo.

El uso de los centros de cómputo tanto en centros educativos, universidades o cualesquiera instituciones ha vuelto indispensable debido a los avances tecnológicos que han cambiado la forma de aprendizaje y trabajo de la humanidad, por ello tener un control automatizado del encendido y apagado de las luminarias; reduce el consumo de energía, haciendo este proyecto viable en economía y conservación del medio ambiente.

Palabras claves: *Prototipo; Arduino; Aplicación móvil; App Inventor*



## **ABSTRACT**

The development of this research work is based on the Internet of Things (IoT) application for the design of a prototype of a lighting control system for computer centers through an application developed in an Android operating system, which is installation in a mobile phone, which is intercommunicated by means of a Bluetooth module, making use of a motion detector sensor that sends an alert signal in the mobile application, with the objective of controlling the power supply for the illumination of a computer center.

The use of computer centers in schools, universities or institutions have become indispensable due to technological advances that have changed the way of learning and the work of humanity, therefore having an automated control of the On and off of the illuminating ones reduce the energy consumption, make this project viable in the economy and the conservation of the environment.

Key words: *Prototype; Arduino; App mobile; App Inventor*

# ÍNDICE GENERAL

1.	Introducción .....	1
2.	Fundamento Teórico .....	2
2.1.	Antecedentes .....	2
2.2.	Marco teórico.....	3
2.2.1.	Arduino Uno .....	3
2.2.2.	Sensor de movimiento .....	4
2.2.3.	Módulo Bluetooth .....	5
2.2.4.	Cables Dupont .....	6
2.2.5.	Protoboard .....	6
2.2.6.	Módulo Relay .....	7
2.2.7.	Aplicación móvil .....	7
2.2.8.	Internet de las cosas.....	8
2.2.9.	App Inventor.....	8
3.	Diseño del prototipo .....	14
3.1.	Gestión de requerimientos .....	14
3.1.1.	Definiciones, acrónimos y abreviaturas .....	14
3.1.2.	Perspectiva del producto. ....	14
3.1.3.	Características de los usuarios .....	14
3.1.4.	Restricciones .....	14
3.1.5.	Suposiciones y dependencias.....	15
3.1.6.	Requerimientos funcionales.....	15
3.1.7.	Requerimientos no funcionales. ....	18
3.1.8.	Diagrama de actores .....	19
3.2.	Diseño de la aplicación móvil.....	24
3.2.1.	Bluetooth.....	27
3.2.2.	Sensor de movimiento PIR .....	28
3.2.3.	Notificaciones Push .....	29
3.2.4.	Botones. ....	29
3.2.5.	Desconexión de Bluetooth.....	30
3.3.	Diseño del circuito.....	30
3.4.	Funcionamiento lógico del prototipo.....	36
3.5.	Código en Arduino.....	38
3.6.	Análisis de costos .....	38

3.7.	Pruebas de Unidad y Pruebas del Sistema .....	39
3.7.1.	Establecer conexión Bluetooth .....	40
3.7.2.	Verificar detección de movimiento .....	41
3.7.3.	Verificar encendido de luminaria .....	42
3.7.4.	Verificar apagado de luminaria .....	43
3.7.5.	Desconectar conexión Bluetooth .....	43
	Conclusiones .....	44
	Recomendaciones .....	45
	Bibliografía .....	46



## ÍNDICE DE TABLAS

Tabla 3. 1 Definiciones, acrónimos .....	14
Tabla 3. 2 Tipo de Usuario – Administrador.....	14
Tabla 3. 3 Requerimiento Funcional: Conexión Bluetooth. ....	15
Tabla 3. 4 Requerimiento Funcional: Detección de movimiento.....	16
Tabla 3. 5 Requerimiento Funcional: Encender luminaria. ....	16
Tabla 3. 6 Requerimiento Funcional: Encender luminaria. ....	17
Tabla 3. 7 Requerimiento Funcional: Desconexión Bluetooth.....	17
Tabla 3. 8 Requerimiento No Funcional: Interfaz de la aplicación.....	18
Tabla 3. 9 Requerimiento no funcional: Disponibilidad de la Aplicación.....	18
Tabla 3. 10 Requerimiento no funcional: Confiabilidad de la Aplicación .....	19
Tabla 3. 11 Caso de uso: Establecer conexión Bluetooth .....	20
Tabla 3. 12 Caso de uso: Verificar detección de movimiento. ....	21
Tabla 3. 13 Caso de uso: Verificar encendido de luminaria. ....	22
Tabla 3. 14 Caso de uso: Verificar apagado de luminaria .....	22
Tabla 3. 15 Caso de uso: Desconectar conexión Bluetooth .....	23
Tabla 3. 16 Materiales de trabajo. ....	38
Tabla 3. 17 Materiales para la elaboración de la maqueta .....	38
Tabla 3. 18 Prueba unitaria: Establecer conexión Bluetooth.....	40
Tabla 3. 19 Prueba unitaria: Verificar detección de movimiento. ....	41
Tabla 3. 20 Prueba unitaria: Verificar encendido de luminaria. ....	42
Tabla 3. 21 Prueba unitaria: Verificar apagado de luminaria.....	43
Tabla 3. 22 Prueba unitaria: Desconectar conexión Bluetooth. ....	43

## ÍNDICE DE FIGURAS

Figura 2. 1 Visión Global de App Inventor.....	9
Figura 2. 2 Diseño Arquitectónico de App Inventor .....	10
Figura 2. 3 Una aplicación puede responder tanto a eventos internos, externos o generador por el usuario.....	12
Figura 3. 1 Diagrama de actores.....	19
Figura 3. 2 Caso de uso: Ingresar al sistema .....	20
Figura 3. 3 Caso de uso: Verificar detección de movimiento.....	21
Figura 3. 4 Caso de uso: Verificar encendido de luminaria.....	21
Figura 3. 5 Caso de uso: Desconectar conexión Bluetooth .....	23
Figura 3. 6 Diseño exterior de la aplicación .....	25
Figura 3. 7 Diagrama de bloques.....	26
Figura 3. 8 Diseño de la aplicación .....	27
Figura 3. 9 Detección de movimiento .....	28
Figura 3. 10 Modelo Notificaciones Push.....	29
Figura 3. 11 Proceso consulta de usuario.....	30
Figura 3. 12 Arduino uno y sus partes.....	31
Figura 3. 13 Módulo PirHC-SR501 .....	32
Figura 3. 14 Modulo Bluetooth HC06.....	33
Figura 3. 15 Módulo Relay 2VH 5VDC .....	34
Figura 3. 16 Conexión de Relay con Bluetooth .....	34
Figura 3. 17 Cable Dupont.....	35
Figura 3. 18 Protoboard 830 .....	35
Figura 3. 19 Foco - Sócate .....	36
Figura 3. 20 Conexiones de los módulos.....	37
Figura 3. 21 Conexión completa.....	37
Figura 3. 22 Prueba unitaria: Establecer conexión Bluetooth.....	40
Figura 3. 23 Prueba unitaria: Verificar detección de movimiento .....	42

# 1. INTRODUCCIÓN

En la actualidad el avance de la tecnología ha permitido el desarrollo de diferentes aplicaciones utilizando dispositivos, donde diferentes actividades cotidianas se han automatizado, facilitando la vida de las personas.

El presente trabajo de investigación “PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL” describe un control del encendido y apagado del sistema de iluminación de los centros de cómputo en función a la detección de presencia de personas por medio de sensores de movimiento. El escenario de aplicación del prototipo está enfocado en los centros de cómputo, para efectos de modelado de la propuesta, se ha tomado como referencia los centros de cómputo de la universidad nacional de Piura. La falta de control o el descuido de parte de los auxiliares de los centros de cómputo, origina que muchas veces las luminarias queden encendidas cuando no están en uso, generando un consumo de energía incensario.

En el desarrollo de este trabajo de investigación se hizo uso de “El Internet de las cosas (IoT)”, ya que actualmente todo lo que nos rodea está conectado a internet y va tomando gran importancia dentro de la vida cotidiana de las personas, esto hace crear nuevas alternativas que hagan uso de esta tendencia, creando soluciones que permitan optimizar muy bien los recursos y el tiempo, además de brindar beneficios a quienes hagan uso de ella.



## 2. FUNDAMENTO TEÓRICO

### 2.1. ANTECEDENTES

El ser humano tiene el tiempo en su contra, el cual muchas veces no realiza actividades que, aunque parezcan prescindibles son de importancia, como el apagar o encender las luces en el momento oportuno, ya que estas permanecen encendidas durante periodos no productivos generando un consumo innecesario de energía eléctrica.

Del mismo modo, para conseguir un ahorro de energía adecuado, es necesario apagar las luces automáticamente al poco tiempo de detectar la falta de presencia de personas que se encontraron en un ambiente cerrado como por ejemplo un centro de cómputo, ya que con este sistema de iluminación se busca facilitar y controlar el ahorro de energía.

De esta forma, se han desarrollado muchos trabajos sobre domótica, de los cuales mencionamos los siguientes por ser de interés:

López, Espinoza, & Barrientos Padilla (2015) en su trabajo **“Implementación de una solución de domótica basado en las mejores soluciones y prácticas del mercado actual”**, desarrollado para optar los títulos de Ingenieros informáticos en la Universidad Peruana de Ciencias Aplicadas, hace un resumen de todas las tecnologías presentes en Perú en el área de la domótica, así también se hace un diseño de un sistema de control de iluminación con una comunicación inalámbrica usando módulos Bluetooth 4.0.

Huamán (2017) en su trabajo **“Control inteligente de sistemas e iluminación en edificios”**, desarrollado para optar el Grado de Magister en Ingeniería Mecánico-Eléctrica con Mención en Automática y Optimización en la Universidad de Piura, busca desarrollar un controlador adaptivo para el control inteligente de la iluminación en edificios, esto considera casas, oficinas o cualquier ambiente donde se requiera iluminar, se va presentar dos opciones de controlador adaptativo, y para

considerarlo un sistema inteligente este se va adaptar al usuario haciendo uso de la tecnología de aprendizaje automático.

García Macías & Marcillo Zambrano (2017) en su trabajo “**Sistema Domótico Mediante Smartphone de la Iluminación en el Auditorio de la Carrera De Computación – ESPAM-MFL**”, desarrollado en la Escuela Superior Politécnica Agropecuaria de Manabí "Manuel Félix López", busca controlar la iluminación del auditorio de la carrera de la computación del mencionado centro de estudios. Para tal fin realizó una investigación documental sobre las implementaciones de sistemas domóticas, su estructura y características que determinaron los elementos y dispositivos requeridos por el sistema e implementar una aplicación móvil que controle la iluminación del auditorio.

## **2.2. MARCO TEÓRICO**

### **2.2.1. Arduino Uno**

Arduino Uno es una board basada en un microcontrolador Atmega328. Tiene 14 pines de entrada/salida digital (de los cuales 4 pueden ser utilizados para salidas PWM), 6 entradas análogas, un resonador cerámico de 16 MHz, un conector para USB tipo hembra, un Jack para fuente de Poder, un conector ICSP y un botón reset. (Naylamp Mechatronics, a)

Tiene todo lo necesario para manejar el controlador, simplemente conectamos al computador por medio del cable USB o una fuente de poder externa, que puede ser un adaptador AC-DC o una batería, cabe aclarar que si se alimenta a través del cable USB en el ordenador no es necesario una fuente externa. (Naylamp Mechatronics, a)

#### 2.2.1.1. Características:

(Naylamp Mechatronics, a)

- **Microcontrolador:** *ATmega328*
- **Voltaje Operativo:** *5v*
- **Voltaje de Entrada (Recomendado):** *7 – 12 v*
- **Pines de Entradas/Salidas Digital:** *14 (De las cuales 6 son salidas PWM)*
- **Pines de Entradas Análogas:** *6*
- **Memoria Flash:** *32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader.*
- **SRAM:** *2 KB (ATmega328)*
- **EEPROM:** *1 KB (ATmega328)*
- **Velocidad del Reloj:** *16 MHZ.*

#### 2.2.2. Sensor de movimiento

Los sensores PIR tienen como función detectar movimiento (de personas), normalmente se busca detectar el movimiento de una persona dentro del rango del sensor. Son conocidos como PIR, “Sensores Infrarrojos” o “Sensores de movimiento”. (Naylamp Mechatronics, b)

Los sensores PIR son perfectos para detectar cuando una persona ingresa o abandona un espacio, a partir de esto podemos programar diferentes acciones como el encendido/apagado automático de luces. (Naylamp Mechatronics, b)

#### **2.2.2.1. Características**

(Naylamp Mechatronics, b)

- Voltaje de operación: 4.5VDC- 20VDC
- Consumo de corriente en reposo: <50uA.
- Voltaje de salida: 3.3V (alto) / 0V (bajo).
- Rango de detección: 3 a 7 metros, ajustable mediante Trimmer (Sx).
- Angulo de detección: <100° (cono)
- Tiempo de retardo: 5-200 S (puede ser ajustado (Tx), por defecto 5S +-3%)
- Tiempo de bloqueo: 2.5 S (por defecto).
- Temperatura de trabajo: -20°C hasta 80°C.

#### **2.2.3. Módulo Bluetooth**

El módulo Bluetooth HC-06 nos permite conectar nuestros proyectos con Arduino a un Smartphone, celular o PC de forma inalámbrica (Bluetooth), con la facilidad de operación de un puerto serial. La transmisión se realiza totalmente en forma transparente al programador, por lo que se conecta en forma directa a los pines seriales de nuestro microcontrolador preferido (respetando los niveles de voltaje, ya que el módulo se alimenta con 3.3V).

(Naylamp Mechatronics, c)

#### **2.2.3.1. Características**

(Naylamp Mechatronics, c)

- Voltaje de Operación: 3.3V - 5VDC
- Corriente de operación: < 40mA
- Corriente modo sleep: < 1mA
- Chip: BC417143I
- Bluetooth: V2.0+EDR

- Frecuencia: Banda ISM de 2.4 GHz
- Potencia de emisión: 4 dBm, clase 2.
- Alcance 10 metros
- Velocidad de transmisión: 1200bps hasta 1.3Mbps.
- Velocidad asincrónica: 2.1Mbps (máx.) / 160 Kbps.
- Compatible con Android
- Seguridad: Autenticación y encriptación.

#### **2.2.4. Cables Dupont**

Conectar tus módulos a tu Arduino ya no será un problema con la ayuda de este magnífico cable dupont. Se entrega en grupos de 20 cables, que se pueden separar para crear "buses" más pequeños. (Naylamp Mechatronics, d)

Este modelo hembra a macho es el más popular entre los cables dupont, pues las placas Arduino ofrecen terminales tipo hembra y la mayoría de modulo terminales tipo macho. Con este cable puede conectar el lado Macho al Arduino y el lado Hembra al módulo o sensor. (Naylamp Mechatronics, d)

##### **2.2.4.1. Características**

(Naylamp Mechatronics, d)

- Tipo: Hembra a Macho
- Espaciado: 2.54mm (0,1 pulgada)
- Longitud: 20cm
- Corriente: 35mA

#### **2.2.5. Protoboard**

El Protoboard 830 sirve para probar circuitos sin tener que fabricar un PCB, cuenta con 4 líneas de energía, 63 columnas y 10 filas para un total de 830 puntos. (Naylamp Mechatronics, e)

Todos los pines están espaciados por un estándar de 0,1". Los dos conjuntos de cinco filas están separados por aproximadamente 0,3", perfecto para integrados tipo DIP. La Junta acepta cables en el rango de 29-20AWG, para mejores resultados utilizar el cable especial para Protoboard. (Naylamp Mechatronics, e)

#### **2.2.5.1. Características**

(Naylamp Mechatronics, e)

- Contactos de bronce fosforoso y níquel plata.
- Vida útil de más 10.000 inserciones.
- plegable y expandible, se pone uno al lado de otro por 3 encastrés.
- compatible con las fuentes para Protoboard.

#### **2.2.6. Módulo Relay**

Dentro de la gran variedad de proyectos que podemos realizar con Arduino, podemos llegar a desear controlar componentes de alto voltaje o alto amperaje, como bombillas o bombas de agua, los cuales no pueden ser manejados directamente con Arduino. En estos casos es necesario utilizar Relay o Reles, estos dispositivos permiten controlar cargas de alto voltaje con una señal pequeña. (Naylamp Mechatronics, f)

Para la programación de Arduino y Relay se recomienda el uso de timers con la función "millis()" y de esa forma no utilizar la función "delay" que impide que el sistema continúe trabajando mientras se activa/desactiva un Relay. (Naylamp Mechatronics, f)

#### **2.2.7. Aplicación móvil**

Una aplicación (también llamada App) es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático. Cabe destacar que, aunque todas las aplicaciones son



programas, no todos los programas son aplicaciones. Existe multitud de software en el mercado, pero sólo se denomina así a aquel que ha sido creado con un fin determinado, para realizar tareas concretas.

#### **2.2.8. Internet de las cosas**

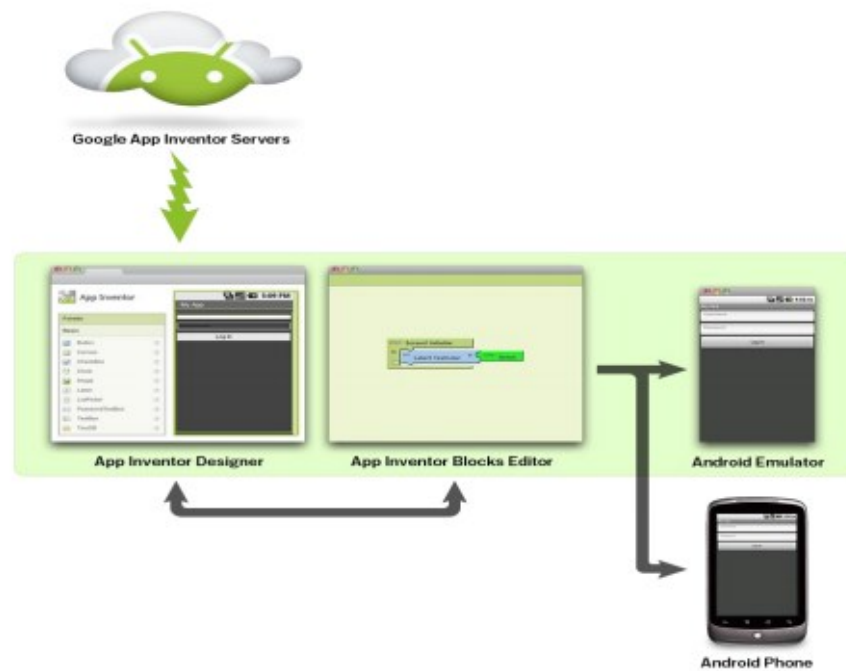
Evans (2011) Indica que Internet de las cosas (IoT), algunas veces denominado "Internet de los objetos", lo cambiará todo, incluso a nosotros mismos. Si bien puede parecer una declaración arriesgada, hay que tener en cuenta el impacto que Internet ha tenido sobre la educación, la comunicación, las empresas, la ciencia, el gobierno y la humanidad. Claramente Internet es una de las creaciones más importantes y poderosas de toda la historia de la humanidad.

El «Internet de las Cosas» (IoT) hace referencia, como se ha adelantado, a una tecnología basada en la conexión de objetos cotidianos a Internet que intercambian, agregan y procesan información sobre su entorno físico para proporcionar servicios de valor añadido a los usuarios finales. También reconoce eventos o cambios, y tales sistemas pueden reaccionar de forma autónoma y adecuada. Su finalidad es, por tanto, brindar una infraestructura que supere la barrera entre los objetos en el mundo físico y su representación en los sistemas de información. (Moises, 2018)

#### **2.2.9. App Inventor**

App Inventor parte de una idea conjunta del Instituto Tecnológico de Massachusetts y de un equipo de Google Education. Se trata de una herramienta web de desarrollo para iniciarse en el mundo de la programación. Con él pueden hacerse aplicaciones muy simples, y también muy elaboradas, que se ejecutarán en los dispositivos móviles con sistema operativo Android. App Inventor es un lenguaje de programación basado en

bloques (como piezas de un juego de construcción), y orientado a eventos. Sirve para indicarle al “cerebro” del dispositivo móvil qué queremos que haga, y cómo. Es por supuesto muy conveniente disponer de un dispositivo Android donde probar los programas según los vamos escribiendo.

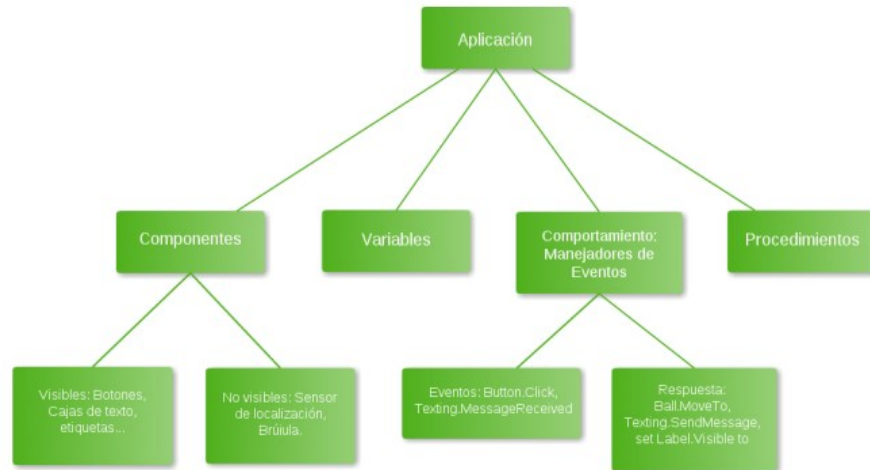


**Figura 2. 1 Visión Global de App Inventor**

Fuente: Universidad de Salamanca (s.f.). BÁSICOS APP INVENTOR: Manual de Introducción a AppInventor. Recuperado de <http://diarium.usal.es/igallego/files/2015/06/Basicos-APPInventor-Manual-de-Introduccion.pdf>

### 2.2.9.1. Diseño arquitectónico

Las aplicaciones construidas mediante App Inventor están compuestas por los elementos que se muestran en el siguiente diagrama:



**Figura 2. 2 Diseño Arquitectónico de App Inventor**

Fuente: Embedded Laboratory (2018) *Getting Started with MQTT using Mosquitto*. Recuperado de

<http://embeddedlaboratory.blogspot.com/2018/01/getting-started-with-mqtt-using.html>

### 2.2.9.2. Componentes de App Inventor

Hay dos tipos de componentes principales en cualquier aplicación: los visibles y los no visibles.

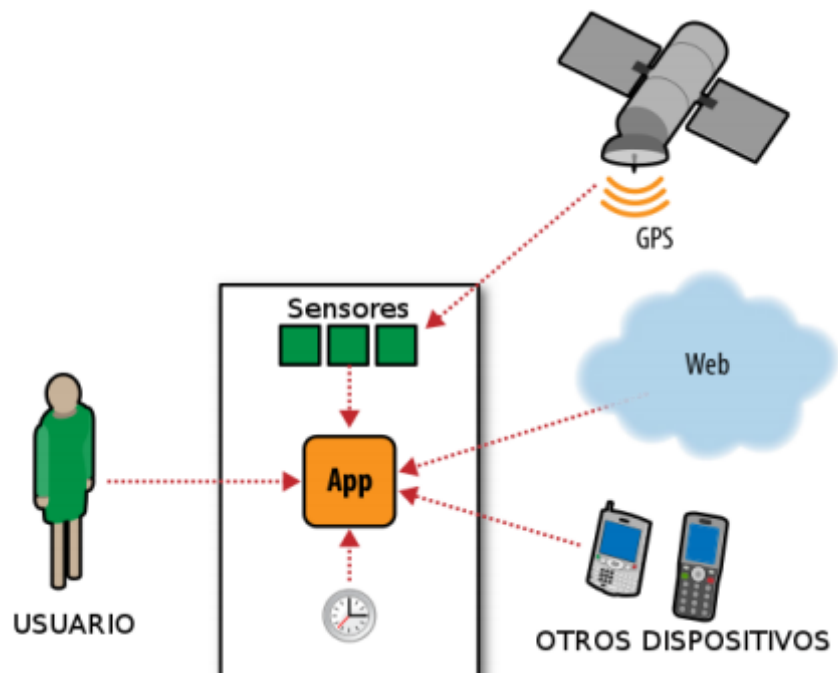
Los componentes visibles son aquellos que podemos ver una vez hemos ejecutado nuestra aplicación (botones, cajas de texto, etiquetas, etc.). El conjunto de estos elementos se denomina comúnmente como la interfaz de usuario de la aplicación.

Por otro lado, los componentes no visibles son aquellos que no podemos ver en la aplicación, ya que no son parte de la interfaz de usuario. Proporcionan acceso a la funcionalidad interna de los dispositivos; por ejemplo, el componente **Texting** permite enviar y procesar mensajes de texto, y el componente **LocationSensor** permite determinar la localización del dispositivo.

Ambos componentes están definidos mediante una serie de propiedades. Las propiedades son fragmentos de memoria que permiten almacenar información relativa al componente al que referencian. Los componentes visibles, por ejemplo, disponen de propiedades relativas a suposición, altura y anchura, y alineación, que definen conjuntamente su aspecto dentro de la aplicación global. Todas estas propiedades se definen dentro del diseñador de componentes de App Inventor.

#### **2.2.9.3. Comportamiento de App Inventor**

AppInventor proporciona un lenguaje visual de bloques que nos permite definir comportamientos de una forma muy precisa. Normalmente, podemos identificar el desarrollo de aplicaciones con la elaboración de una “receta”, es decir, siguiendo una secuencia lineal de instrucciones. La mayoría de las aplicaciones actuales no cumple estrictamente este tipo de paradigma. No se ejecutan una serie de instrucciones en un orden predeterminado, sino que, la aplicación reacciona a una serie de eventos, normalmente iniciados por el usuario final de la aplicación. Por ejemplo, si el usuario hace clic sobre un botón, la aplicación responde realizando alguna operación (enviar un mensaje de texto, confirmar una determinada operación, etc.). Este tipo de aplicaciones se pueden interpretar como un conjunto de componentes que reaccionan ante unos determinados eventos. Las aplicaciones incluyen una serie de “recetas” (secuencias de instrucciones), las cuales se ejecutan cuando se producen los eventos asociados a las mismas.



**Figura 2. 3 Una aplicación puede responder tanto a eventos internos, externos o generados por el usuario.**

**Fuente:** Universidad de Salamanca (s.f.). BÁSICOS APP INVENTOR: Manual de Introducción a AppInventor. Recuperado de <http://diarium.usal.es/igallego/files/2015/06/Basicos-APPInventor-Manual-de-Introduccion.pdf>

El motivo principal por el cual podemos afirmar que las aplicaciones desarrolladas mediante AppInventor son intuitivas, es porque están basadas en el paradigma evento-respuesta.

#### 2.2.9.4. Eventos

Las aplicaciones creadas con App Inventor están orientadas a eventos. No llevan a cabo un conjunto de instrucciones en un orden predeterminado, sino que reaccionan a eventos. Al hacer clic en un botón, arrastrar el dedo o tocar en la pantalla son eventos. Con App Inventor, toda la actividad se produce en respuesta a un evento. Tu aplicación no debe contener bloques fuera de una pieza "when do" de un evento.

Como se producen un evento, la aplicación reacciona llamando a una secuencia de instrucciones como establecer el color de fondo de un botón a azul o cambiar el texto de una etiqueta.

Los eventos pueden ser divididos en 2 tipos diferentes: automáticos e iniciados por el usuario. Hacer clic en un botón, tocar o arrastrar en la pantalla, inclinar el teléfono son eventos iniciados por el usuario. Una colisión entre Sprites o de estos con los bordes de un Canvas son eventos automáticos.

### 3. DISEÑO DEL PROTOTIPO

#### 3.1. GESTIÓN DE REQUERIMIENTOS

En esta sección se definen los requisitos funcionales y no funcionales que se tienen que cumplir para que la aplicación móvil funcione de manera correcta. Es necesario registrar estos requisitos mediante diagramas de casos de uso.

##### 3.1.1. Definiciones, acrónimos y abreviaturas

**Tabla 3. 1 Definiciones, acrónimos**

<b>Nombre</b>	<b>Descripción</b>
<b>Usuario</b>	Persona que usará la aplicación móvil.
<b>SISMOT</b>	SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL.
<b>RF</b>	Requerimiento Funcional
<b>RNF</b>	Requerimiento No Funcional

*Fuente: Elaboración propia*

##### 3.1.2. Perspectiva del producto.

La aplicación móvil será diseñada y desarrollada para trabajar en entorno móvil lo que permitirá un acceso más fácil y amigable para el usuario, esto con el fin de que haga uso de todas las funcionalidades que posea el aplicativo.

##### 3.1.3. Características de los usuarios

**Tabla 3. 2 Tipo de Usuario – Administrador**

<b>Tipo de usuario</b>	Administrador
<b>Formación</b>	Centro de computo
<b>Actividades</b>	Control de todas las opciones del sistema

*Fuente: Elaboración propia*

##### 3.1.4. Restricciones

- La aplicación establecerá un protocolo de comunicación Bluetooth.
- La aplicación será capaz de enviar y recibir datos concurrentemente.
- La aplicación seguirá un diseño de componentes electrónicos (Hardware, Firmware y Software).



- La aplicación funcionará en dispositivos móviles con plataforma Android.
- La aplicación deberá tener un diseño de interfaz sencilla y amigable al usuario.

### 3.1.5. Suposiciones y dependencias

Se supone que los usuarios tienen conocimientos básicos sobre el uso de celular. Los equipos deben ser estables y cumplir con los requisitos indicados.

### 3.1.6. Requerimientos funcionales

#### 3.1.6.1. Conexión Bluetooth.

Establece la comunicación entre el dispositivo móvil que tiene instalada la aplicación y el Arduino Uno, para el envío y recepción de datos.

**Tabla 3. 3 Requerimiento Funcional: Conexión Bluetooth.**

<b>Identificación del requerimiento:</b>	RF01
<b>Nombre del Requerimiento:</b>	Conexión Bluetooth.
<b>Características:</b>	Se podrá establecer conexión Bluetooth.
<b>Descripción del requerimiento:</b>	La aplicación permitirá establecer comunicación Bluetooth con el Arduino Uno.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> </ul>
<b>Prioridad del requerimiento:</b> Alta	

*Fuente: Elaboración propia*

### 3.1.6.2. Detección de movimiento

El usuario podrá visualizar en la aplicación si hay o no movimiento en el centro de cómputo en tiempo real, y también recibirá notificaciones de aviso sobre la detección del movimiento.

**Tabla 3. 4 Requerimiento Funcional: Detección de movimiento**

<b>Identificación del requerimiento:</b>	RF02
<b>Nombre del Requerimiento:</b>	Detección de movimiento.
<b>Características:</b>	Se podrá verificar y notificar si hay o no movimiento dentro del centro de cómputo.
<b>Descripción del requerimiento:</b>	El usuario podrá visualizar en la aplicación si hay o no movimiento en el centro de cómputo y recibirá notificaciones de aviso sobre la detección del movimiento.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"><li>• RNF01</li><li>• RNF02</li><li>• RNF03</li></ul>
<b>Prioridad del requerimiento:</b> Alta	

*Fuente: Elaboración propia*

### 3.1.6.3. Encender luminaria.

El usuario podrá encender la luminaria del centro de cómputo mediante la aplicación móvil haciendo un clic.

**Tabla 3. 5 Requerimiento Funcional: Encender luminaria.**

<b>Identificación del requerimiento:</b>	RF03
<b>Nombre del Requerimiento:</b>	Encender luminaria.
<b>Características:</b>	Se podrá encender la luminaria del centro de cómputo.
<b>Descripción del requerimiento:</b>	La aplicación permitirá al usuario encender la luminaria del centro de cómputo.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"><li>• RNF01</li><li>• RNF02</li><li>• RNF03</li></ul>
<b>Prioridad del requerimiento:</b> Alta	

*Fuente: Elaboración propia*

#### 3.1.6.4. Apagar luminaria.

El usuario podrá apagar la luminaria del centro de cómputo mediante la aplicación móvil haciendo un clic.

**Tabla 3. 6 Requerimiento Funcional: Encender luminaria.**

<b>Identificación del requerimiento:</b>	RF04
<b>Nombre del Requerimiento:</b>	Apagar luminaria.
<b>Características:</b>	Se podrá apagar la luminaria del centro de cómputo.
<b>Descripción del requerimiento:</b>	La aplicación permitirá al usuario apagar la luminaria del centro de cómputo.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"><li>• RNF01</li><li>• RNF02</li><li>• RNF03</li></ul>
<b>Prioridad del requerimiento:</b> Alta	

*Fuente: Elaboración propia*

#### 3.1.6.5. Desconexión Bluetooth.

El usuario podrá desconectar la comunicación Bluetooth entre el dispositivo móvil que tiene instalada la aplicación y el Arduino Uno, haciendo un clic.

**Tabla 3. 7 Requerimiento Funcional: Desconexión Bluetooth**

<b>Identificación del requerimiento:</b>	RF05
<b>Nombre del Requerimiento:</b>	Desconexión Bluetooth.
<b>Características:</b>	Se podrá desconectar la conexión Bluetooth.
<b>Descripción del requerimiento:</b>	La aplicación permitirá al usuario desconectar la comunicación Bluetooth con el Arduino Uno.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"><li>• RNF01</li><li>• RNF02</li><li>• RNF03</li></ul>
<b>Prioridad del requerimiento:</b> Alta	

*Fuente: Elaboración propia*

### 3.1.7. Requerimientos no funcionales.

#### 3.1.7.1. Interfaz de la aplicación.

La interfaz de la aplicación debe ser amigable y sencilla para que el usuario pueda interactuar teniendo una experiencia de manejo fácil.

**Tabla 3. 8 Requerimiento No Funcional: Interfaz de la aplicación**

<b>Identificación del requerimiento:</b>	RNF01
<b>Nombre del Requerimiento:</b>	Interfaz del sistema.
<b>Características:</b>	La aplicación presentara una interfaz sencilla para que sea de fácil manejo a los usuarios de la misma.
<b>Descripción del requerimiento:</b>	El sistema debe tener una interfaz de uso intuitiva y sencilla.
<b>Prioridad del requerimiento:</b> Alta	

*Fuente: Elaboración propia*

#### 3.1.7.2. Disponibilidad de la aplicación.

La aplicación móvil, para el control de iluminación del centro de cómputo, debe funcionar en cualquier momento y por el tiempo que lo requiera el usuario.

**Tabla 3. 9 Requerimiento no funcional: Disponibilidad de la Aplicación**

<b>Identificación del requerimiento:</b>	RNF02
<b>Nombre del Requerimiento:</b>	Disponibilidad de la Aplicación.
<b>Características:</b>	La Aplicación Móvil podrá funcionar a cualquier hora y durante las 24 horas del día, para el control de iluminación de los centros de cómputo.
<b>Descripción del requerimiento:</b>	La aplicación móvil tendrá disponibilidad de usarse en cualquier momento, mediante conexión Bluetooth. Esto dependerá del usuario.
<b>Prioridad del requerimiento:</b> Alta	

*Fuente: Elaboración propia*

### 3.1.7.3. Confiabilidad de la Aplicación.

La aplicación móvil debe enviar y recibir datos confiables al Arduino Uno, mediante comunicación Bluetooth, para el control de iluminación del centro de cómputo.

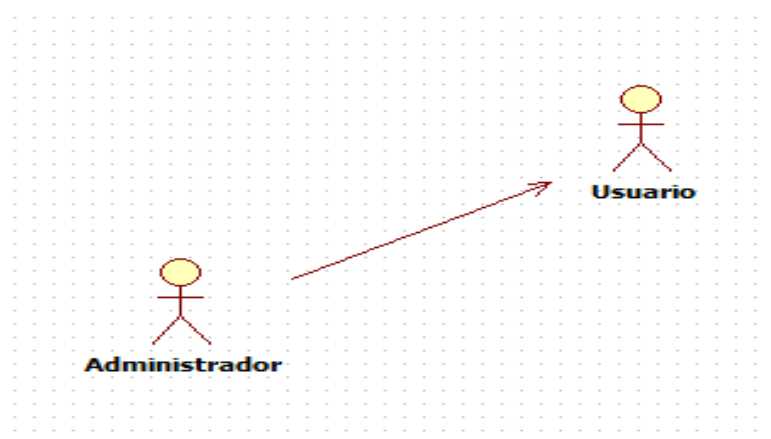
**Tabla 3. 10 Requerimiento no funcional: Confiabilidad de la Aplicación**

<b>Identificación del requerimiento:</b>	RNF03
<b>Nombre del Requerimiento:</b>	Confiabilidad de la Aplicación.
<b>Características:</b>	La Aplicación Móvil mantendrá comunicación Bluetooth confiable, para el control de iluminación de los centros de cómputo.
<b>Descripción del requerimiento:</b>	La aplicación móvil durante la conexión Bluetooth establecida, debe enviar y recibir datos de manera real y confiable con los componentes del sistema de circuitos electrónicos, para el control de iluminación del centro de cómputo.
<b>Prioridad del requerimiento:</b>	Alta

*Fuente: Elaboración propia*

### 3.1.8. Diagrama de actores

En esta sección se define los casos de uso, para especificar y describir la comunicación y comportamiento de la aplicación móvil mediante la interacción con el usuario.



**Figura 3. 1 Diagrama de actores**

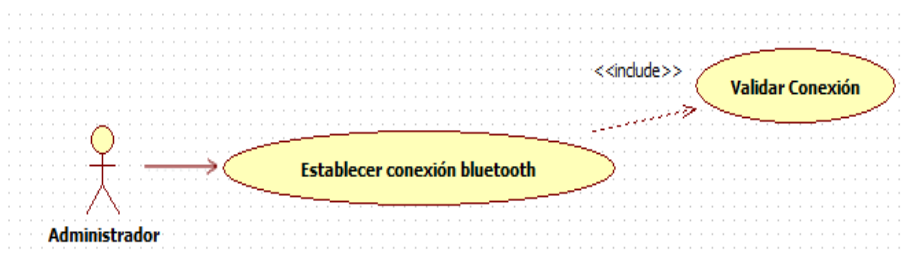
*Fuente: Elaboración propia*

### 3.1.8.1. Descripción de los actores

**Nombre del actor:** ADMINISTRADOR

**Descripción:** Es el operador o la persona a cargo de controlar la iluminación del centro de cómputo, mediante la aplicación móvil.

### 3.1.8.2. Descripción de casos de uso



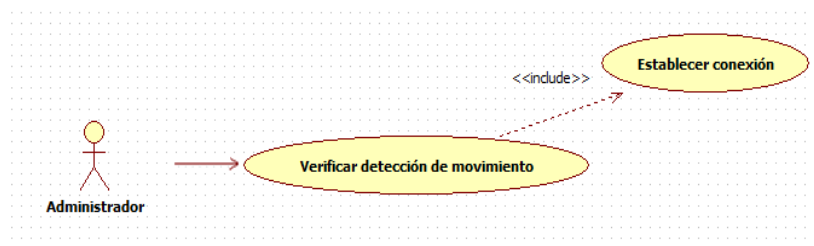
**Figura 3. 2 Caso de uso: Ingresar al sistema**

*Fuente: Elaboración propia*

**Tabla 3. 11 Caso de uso: Establecer conexión Bluetooth**

DESCRIPCIÓN CASO DE USO: ESTABLECER CONEXIÓN BLUETOOTH		
DESCRIPCIÓN	Para poder establecer conexión Bluetooth, es necesario validar los datos de conexión con el Arduino Uno (sistema de componentes electrónicos).	
ACTORES	Administrador	
PRE – CONDICIONES	Los usuarios deben activar la conexión Bluetooth del dispositivo móvil.	
POST – CONDICIONES	Autenticación de conexión exitosa.	
FLUJO PRINCIPAL		
N°	ACTOR	APLICACIÓN
1	Clic en botón conectar.	El sistema muestra lista de conexiones Bluetooth disponibles.
2	El usuario elige la conexión Bluetooth del Arduino UNO.	La aplicación verifica si la conexión es válida.
3		La aplicación muestra la interfaz principal con las opciones correspondientes al usuario.
Excepciones:		
e1.- No hace clic al botón “Ingresar”.		
e2.- Error en la conexión. Mostrar mensaje de error		

*Fuente: Elaboración propia*

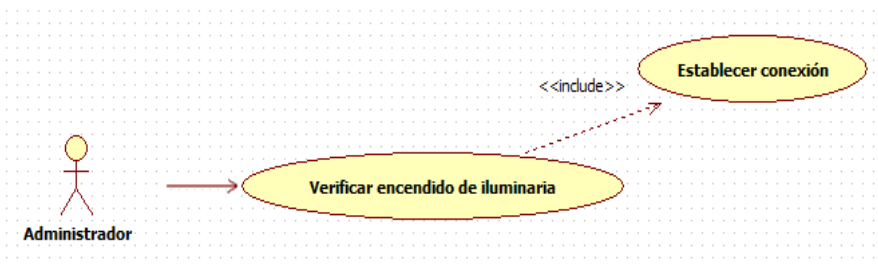


**Figura 3. 3 Caso de uso: Verificar detección de movimiento**  
 Fuente: Elaboración propia

**Tabla 3. 12 Caso de uso: Verificar detección de movimiento.**

DESCRIPCIÓN CASO DE USO: Verificar detección de movimiento.		
DESCRIPCIÓN	Permite verificar si hay o no movimiento en tiempo real dentro del centro de cómputo.	
ACTORES	Administrador	
PRE – CONDICIONES	Conexión Bluetooth establecida	
POST – CONDICIONES		
FLUJO PRINCIPAL		
N°	ACTOR	APLICACIÓN
1	Ingresar a la aplicación.	La aplicación envía notificaciones emitiendo mensaje y sonido de detección de movimiento y muestra en la interfaz principal un botón de color verde con texto “ <b>Hay movimiento</b> ”, cuando se detecta movimiento en el centro de cómputo y un botón de color rojo con texto “ <b>No hay movimiento</b> ”, cuando no se detecta movimiento en el centro de cómputo.
Excepciones:		
e1.- La aplicación pierde la conexión Bluetooth. Mensaje de error.		

Fuente: Elaboración propia



**Figura 3. 4 Caso de uso: Verificar encendido de luminaria**  
 Fuente: Elaboración propia.



**Tabla 3. 13 Caso de uso: Verificar encendido de luminaria.**

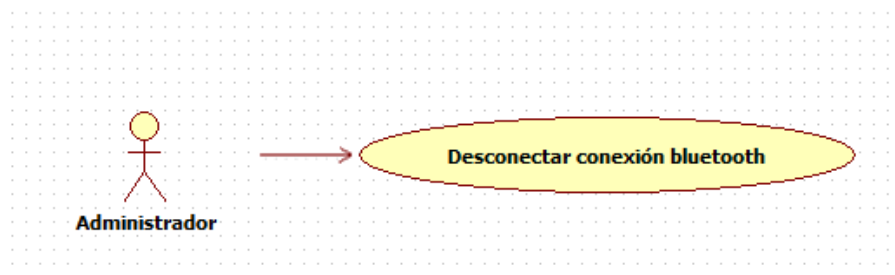
DESCRIPCIÓN CASO DE USO: Verificar encendido de luminaria.		
DESCRIPCIÓN	Permite verificar si la aplicación enciende luminaria del centro de cómputo.	
ACTORES	Administrador	
PRE – CONDICIONES	Conexión Bluetooth establecida	
POST – CONDICIONES		
FLUJO PRINCIPAL		
Nº	ACTOR	APLICACIÓN
1	Hacer clic en botón “Encender”.	La aplicación envía la orden de encender la luminaria del centro de cómputo, mediante la comunicación Bluetooth establecida con el Arduino Uno.
Excepciones: e1.- La aplicación pierde la conexión Bluetooth. Mensaje de error.		

*Fuente: Elaboración propia*

**Tabla 3. 14 Caso de uso: Verificar apagado de luminaria**

DESCRIPCIÓN CASO DE USO: Verificar apagado de luminaria.		
DESCRIPCIÓN	Permite verificar que la aplicación apague luminaria del centro de cómputo.	
ACTORES	Administrador	
PRE – CONDICIONES	Conexión Bluetooth establecida	
POST – CONDICIONES		
FLUJO PRINCIPAL		
Nº	ACTOR	SISTEMA
1	Hacer clic en botón “Apagar”.	La aplicación debe enviar la orden de apagar la luminaria del centro de cómputo, mediante la comunicación Bluetooth establecida con el Arduino Uno.
Excepciones: e1.- La aplicación pierde la conexión Bluetooth. Mensaje de error.		

*Fuente: Elaboración propia*



**Figura 3. 5 Caso de uso: Desconectar conexión Bluetooth**  
*Fuente: Elaboración propia*

**Tabla 3. 15 Caso de uso: Desconectar conexión Bluetooth**

DESCRIPCIÓN CASO DE USO: Desconectar conexión Bluetooth		
<b>DESCRIPCIÓN</b>		Desconectar conexión Bluetooth
<b>ACTORES</b>		El usuario podrá desconectar la conexión Bluetooth, con el Arduino Uno (sistema de componentes electrónicos).
<b>PRE – CONDICIONES</b>		
<b>POST – CONDICIONES</b>		
<b>FLUJO PRINCIPAL</b>		
Nº	ACTOR	SISTEMA
1	Clic en botón “Desconectar”.	Desconecta la conexión Bluetooth de la aplicación con el Arduino Uno (sistema de componentes electrónicos)
<b>Excepciones:</b> <b>Ninguna.</b>		

*Fuente: Elaboración propia*

### 3.2. DISEÑO DE LA APLICACIÓN MÓVIL

Se crea un nuevo proyecto en App inventor: Aplicación Móvil. Después, se construye el diseño de la App. En el centro se visualiza una pantalla de un dispositivo Android, donde se irán poniendo todos los componentes que se usarán en el diseño de la aplicación. Estos componentes se encuentran en la paleta izquierda, dividida por secciones.

El diseño de la App o aplicación consta de una pantalla o interfaz, para la cual se utiliza las siguientes secciones de componentes:

- Sección, interfaz de usuario: botones, imagen, etiquetas y selector de lista.
- Sección, disposición: paneles de orientación horizontal y vertical; para ubicar a los componentes de interfaz de usuario en la posición deseada.
- Sección Medios: componente de sonido, para emitir un sonido cuando se envíe una notificación al celular.
- Sección, sensores: reloj, el cual permitirá establecer el tiempo (milisegundos), en que la aplicación recibirá datos mediante la comunicación Bluetooth con el módulo Arduino UNO.
- Sección, conectividad: componente **CienteBluetooth**, para establecer conexión/comunicación, entre el módulo Arduino UNO y un cliente Bluetooth.
- Sección Extensión: Se importa la extensión: **com.puravidaapps.TaifunNotification.aix**, la cual permite enviar notificaciones **Push** al celular, para avisar al usuario cuando hay o no movimiento.

Las propiedades de cada componente se usarán para cambiar colores, medidas, tamaños, tipos de letras, etc.; para mejorar el diseño de la aplicación.

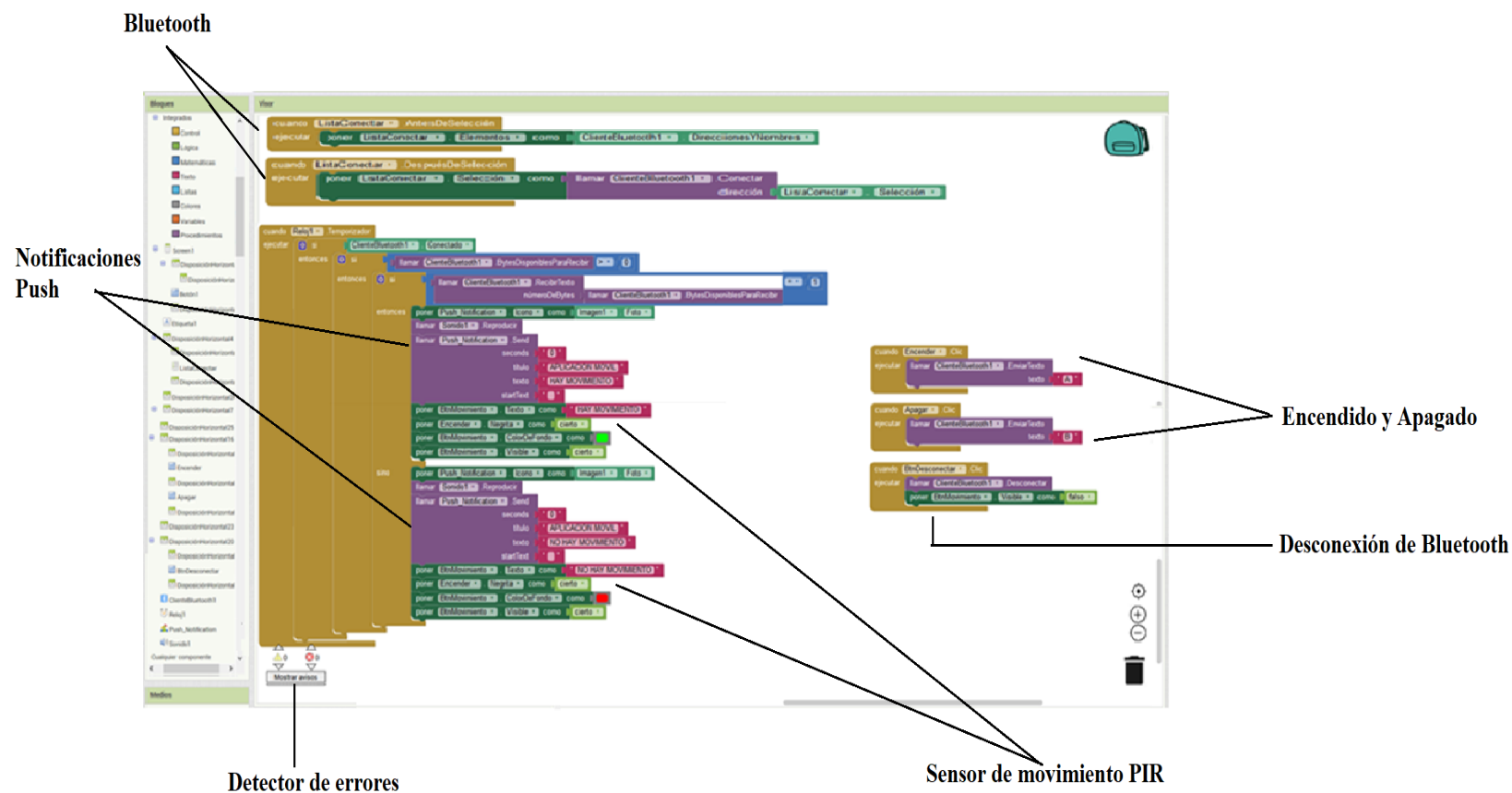


**Figura 3. 6 Diseño exterior de la aplicación**  
Fuente: Elaboración propia

Terminado el diseño de la App. Se programará la aplicación en forma de bloques.

Se muestra una paleta izquierda, la cual está dividida por secciones: bloques y medios.

La sección bloques, contiene los elementos integrados y las funciones, tales como sentencias de control, funciones lógicas, funciones matemáticas, strings, etc. También se encuentran los elementos y las funciones específicas de los componentes que se insertaron en el diseño de la aplicación.



**Figura 3. 7 Diagrama de bloques**  
 Fuente: Elaboración propia

### 3.2.1. Bluetooth

La comunicación Bluetooth se establece mediante el selector de lista Conectar. Antes de pulsarlo, si el Bluetooth del dispositivo Android está listo, la aplicación recogerá todas las conexiones Bluetooth listadas en este. Esto lo hará al principio de abrir la App. Al pulsarlo, muestra un listado, donde se selecciona el módulo Bluetooth conectado al Arduino UNO. Ver **figura 3.8**



**Figura 3. 8** Diseño de la aplicación  
*Fuente: Elaboración propia*

### 3.2.2. Sensor de movimiento PIR

La aplicación recibe la detección de movimiento, enviada por el sensor PIR, mediante la comunicación Bluetooth y muestra un botón de color verde con el texto: “**HAY MOVIMIENTO**”, y un botón de color rojo con el texto: “**NO HAY MOVIMIENTO**”, según sea el caso. Ver **figura 3.9**.

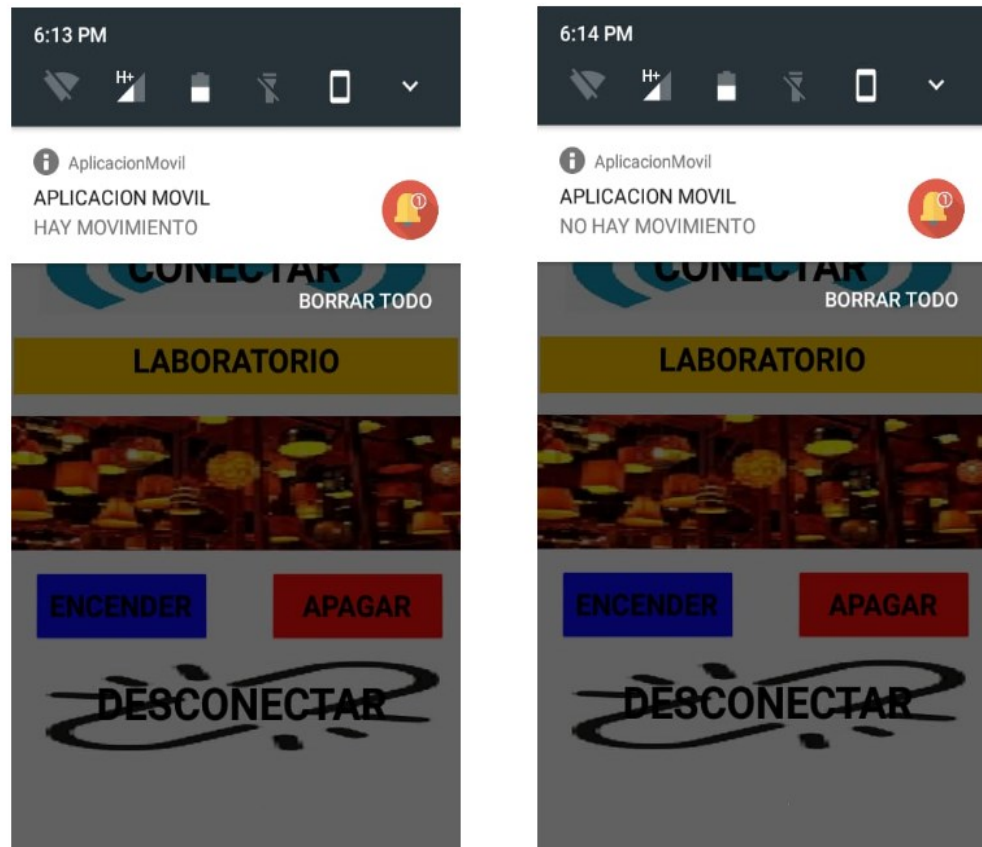


**Figura 3. 9 Detección de movimiento**  
*Fuente: Elaboración propia*



### 3.2.3. Notificaciones Push

La aplicación envía notificaciones de aviso, emitiendo sonido y mensaje “Hay movimiento” o “No hay movimiento” a la barra de deslizamiento superior del celular, cuando recibe la detección de movimiento. Ver **figura 3.10**.



**Figura 3. 10 Modelo Notificaciones Push**  
*Fuente: Elaboración propia*

### 3.2.4. Botones.

Los botones detectan clics, cuya funcionalidad es encender y apagar la luminaria del centro de cómputo. Esta aplicación está diseñada para controlar solo una luminaria.

### 3.2.5. Desconexión de Bluetooth.

El botón desconectar, desactiva la conexión vía Bluetooth de la aplicación con el módulo Arduino UNO. Ver **figura 3.11**.



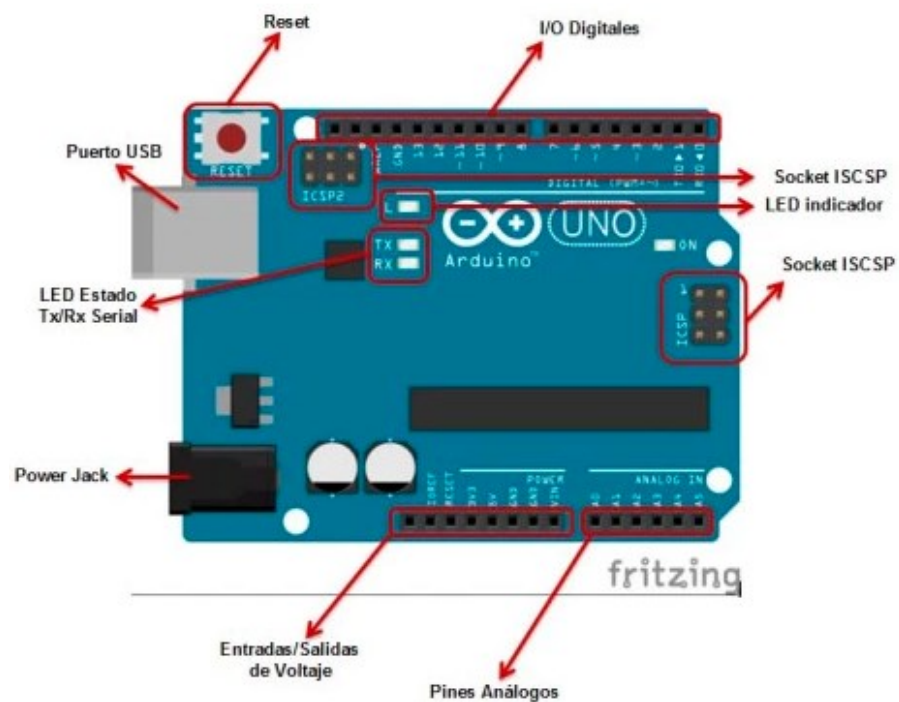
**Figura 3. 11 Proceso consulta de usuario**  
*Fuente: Elaboración propia*

### 3.3. DISEÑO DEL CIRCUITO

Describiremos teóricamente cada dispositivo utilizado, así como su uso específico para el desarrollo de este prototipo y su importancia en la codificación.

Como microcontrolador, se decidió utilizar la placa de ARDUINO UNO ATMEGA328, ya que contamos con la ventaja de haber trabajado anteriormente con esta placa, además de ser de bajo costo, ser de plataforma abierta y fácil de

usar ARDUINO UNO ATMEGA328 está desarrollado por Arduino.cc. La placa está equipada con varios pines digitales y analógicos de entrada/salida que pueden estar interconectados a otros circuitos. Para nuestro prototipo de sistema de control de iluminación se requiere de este dispositivo el cual hace la función del cerebro de nuestro sistema, tiene un conector USB adaptado a través del cual subimos nuestro programa. Se utilizaron los pines digitales 7 y 9, para la transmisión de señal del sensor PIR y el relé respectivamente, se utilizó el GND — puerto tierra y el VCC— puerto positivo, así como los pines digitales 0 y 1, los cuales se conectarán con el Bluetooth, para la recepción y transmisión de datos respectivamente.



**Figura 3. 12 Arduino uno y sus partes**

Fuente: Guerrero (2014). *Arduino Uno: Especificaciones y características*. Recuperado de <https://pluselectric.wordpress.com/2014/09/21/arduino-uno-especificaciones-y-caracteristicas/>

Para la detección de movimiento hacemos uso del SENSOR INFRARROJO PASIVO (PIR). Este dispositivo permite la detección de movimiento, se basa en la medición de la radiación infrarroja de los cuerpos. De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica, que es captada por el sensor y emite una señal digital. Hacemos uso de este dispositivo, el cual al detectar movimiento en el centro de cómputo envía “1” a la aplicación y muestra un mensaje de color verde que indica “Hay movimiento” y cuando no se detecta movimiento envía “0” y muestra en la aplicación un mensaje de color rojo que indica “No hay movimiento”. El sensor infrarrojo PIR posee 3 pines tierra-positivo y el pin de señal los cuales se muestran en la imagen a continuación, como se mencionó anteriormente en pin de señal se declarará en el pin digital 7 del Arduino, el puerto tierra y positivo-VCC con el pin tierra y positivo del Arduino o del Protoboard.

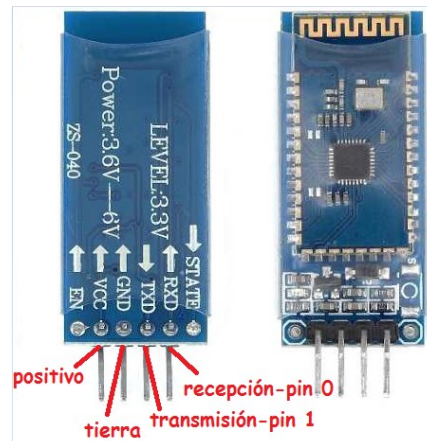


**Figura 3. 13 Módulo PirHC-SR501**

*Fuente:* Naylamp Mechatronics (b). *Módulo PIR HC-SR501*. Recuperado de <https://naylampmechatronics.com/sensores-proximidad/55-modulo-de-deteccion-pir-hc-sr501.html>

Para que se establezca la comunicación entre el Arduino y la aplicación móvil hacemos uso del MÓDULO BLUETOOTH HC – 06 este dispositivo permite la comunicación inalámbrica con otros dispositivos, se comporta como esclavo, esperando peticiones de conexión. Si algún dispositivo se conecta, el HC-06 transmite a este todos los datos que recibe del Arduino y viceversa. Para que se

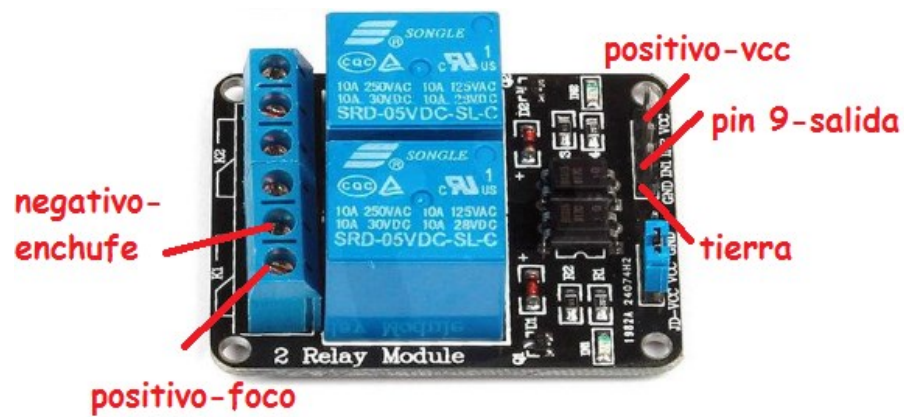
pueda establecer la comunicación entre la aplicación móvil y el Arduino hacemos uso de la librería serial, la cual permite recibir datos a través del comando `serial.read` y enviar datos a través del comando `serial.write`.



**Figura 3. 14 Módulo Bluetooth HC06**

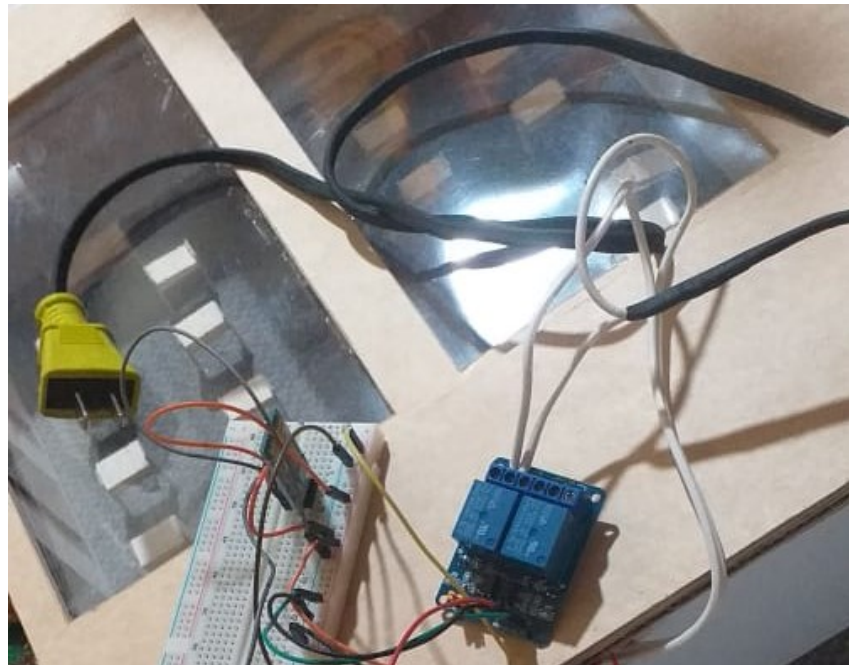
Fuente: Naylamp Mechatronics (c). Módulo Bluetooth HC06. Recuperado de <https://naylampmechatronics.com/inalambrico/24-modulo-bluetooth-hc06.html>

Para el control de la corriente alterna y así proporciona iluminación a los centros de cómputo en el momento requerido se necesita de un dispositivo electromagnético. Es así que decidimos hacer uso del MODULO RELÉ/RELAY, este funciona como un interruptor controlado por un circuito eléctrico. Estos dispositivos permiten controlar cargas de alto voltaje con una señal pequeña. El uso de este dispositivo para la implementación de nuestro circuito se lleva a cabo con la conexión al foco, a la corriente, al Arduino y al Protoboard. Este dispositivo es el que permite que la luminaria se encienda o se apague.



**Figura 3. 15 Módulo Relay 2VH 5VDC**

Fuente: Naylamp Mechatronics (f) Módulo Relay. Recuperado de <https://naylampmechatronics.com/drivers/31-modulo-relay-2-canales-5vdc.html>

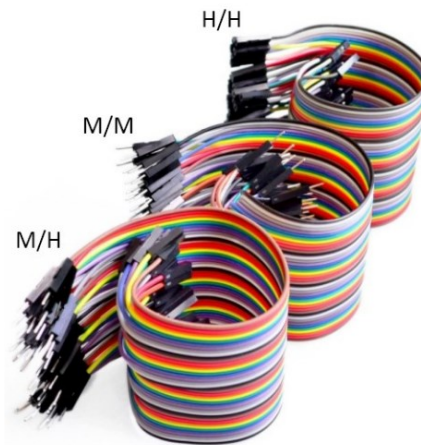


**Figura 3. 16 Conexión de Relay con Bluetooth**

Fuente: Elaboración propia

Para la conexión entre dispositivos, utilizamos los Cables Jumpers O Dupont, los cuales facilitan la conexión en prototipos, sensores y otros dispositivos electrónicos. Existen cables con 3 terminales diferentes, macho-macho, hembra-hembra y macho-hembra. Para el desarrollo del prototipo que implementamos se hace uso de los 3 dependiendo los pines del dispositivo y la distancia requerida entre estos.

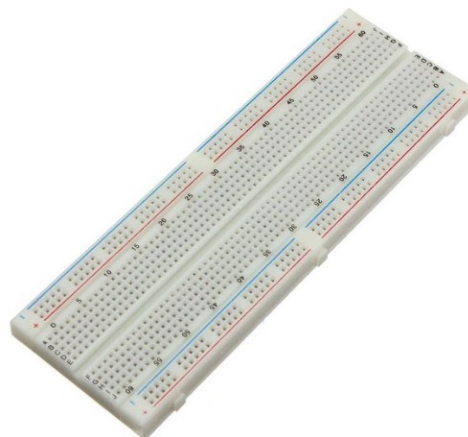




**Figura 3. 17 Cable Dupont**

Fuente: Naylamp Mechatronics (d) Cable Dupont hembra a macho 20cm / 20Und.  
 Recuperado de: <https://naylampmechatronics.com/accesorios-y-prototipado/81-cable-dupont-hembra-a-macho-20cm-x-20und.html>

Para el diseño del circuito necesitábamos un dispositivo el cual no facilite más puntos o puertos GND y tierra es por ellos que hacemos uso del PROTOBOARD 830, este es un tablero con orificios, en la cual se pueden insertar componentes electrónicos y cables para armar circuitos.



**Figura 3. 18 Protoboard 830**

Fuente: Naylamp Mechatronics (e) Protoboard. Obtenido de Naylamp Mechatronics:  
<https://naylampmechatronics.com/accesorios-y-prototipado/14-protoboard-830.html>

Foco – Sócate - Conductor de cordón (cable para la conexión eléctrica); en el desarrollo de nuestro prototipo se hizo uso de un foco para demostrar el funcionamiento y el control de la luminaria, para ello se realizó la conexión con

el sócate y con el enchufe a través del conductor cordón, el cual se conecta al módulo relé y de este a la corriente.



**Figura 3. 19 Foco - Sócate**  
*Fuente: Elaboración propia*

### **3.4. FUNCIONAMIENTO LÓGICO DEL PROTOTIPO**

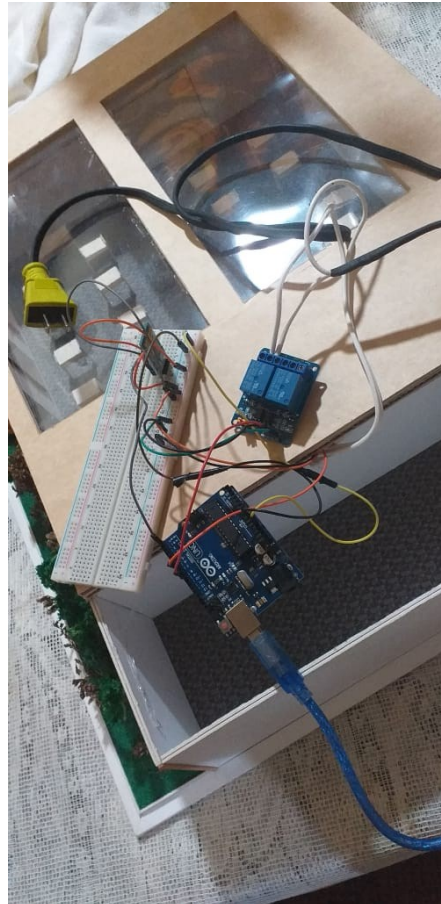
La comunicación entre el ARDUINO y la aplicación móvil es a través de un módulo BLUETOOTH HC 06.

El usuario de la aplicación es quien envía a través de la misma la orden de apagar y encender las luces, teniendo en cuenta si existe o no movimiento en los centros de cómputo, el cual será detectado por el sensor de movimiento (PIR).

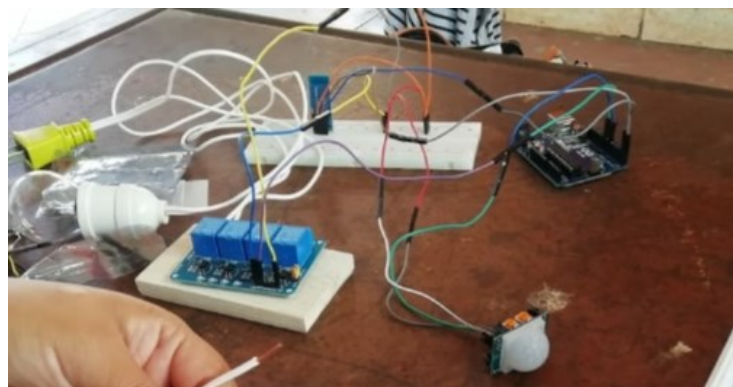
El usuario de la app será alertado a través de notificaciones que llegan directamente a su celular, las cuales indican “Hay Movimiento”/“No hay movimiento”; de no existir movimiento y si las luces se encuentran encendidas pasado 30 segundos éstas automáticamente se apagaran, si se detecta movimiento y no se recibe la orden de encender las luces, estas se encenderán automáticamente pasado 30 segundos, automatizando así los procesos, con el fin del control absoluto de la luminaria en los centros de cómputo, generando un ahorro de energía, ya que la luminaria solo permanecerá encendida en los tiempos necesarios.



Tras incluir los componentes citados anteriormente en el código del proyecto e implementando la solución para que cada uno de ellos realizara la funcionalidad deseada, llegamos a nuestro circuito final, que es el que se muestra en las siguientes imágenes.



**Figura 3. 20 Conexiones de los módulos**  
*Fuente: Elaboración propia*



**Figura 3. 21 Conexión completa**  
*Fuente: Elaboración propia*

### 3.5. CÓDIGO EN ARDUINO

En este trabajo de investigación se utilizó el lenguaje de programación Arduino, este le indica al microcontrolador que acciones realizar cuando esté en funcionamiento.

En la codificación se ha importado la librería SoftwareSerial.h.

Se han utilizado las funciones void setup(), donde se declaran las variables de entrada, salida y el puerto serial y la función void loop () donde se implementa el código que permite apagar/encender las luminarias del centro de cómputo.

También se utilizó la función millis(), la cual permite determinar el tiempo real que lleva en ejecución el programa. En este trabajo se ha implementado un tiempo de 30 segundos para el apagado automático de las luminarias del centro de cómputo.

Ver **Anexo N° 01**.

### 3.6. ANÁLISIS DE COSTOS

Para el desarrollo de la implementación del trabajo de investigación se verificó inventario en base a los costos por herramientas, materiales de trabajo a través de las facturas.

**Tabla 3. 16 Materiales de trabajo.**

N°	Materiales de trabajo	cantidad	Precio unitario	subtotal	IGV	Total
1	Modulo relay	1	S/. 30.00	S/. 30.00	S/. 5.40	S/. 35.40
2	Sensor de movimiento	1	S/. 8.00	S/. 8.00	S/. 1.44	S/. 9.44
3	Cable DUPONT HEMBRA A MACHO 20CM / 20UND	15	S/. 12.00	S/. 12.00	S/. 2.16	S/. 14.16
4	Placa arduino uno	1	S/. 40.00	S/. 40.00	S/. 7.20	S/. 47.20
5	Protoboard	1	S/. 40.00	S/. 40.00	S/. 7.20	S/. 47.20
6	Modulo bluetooth	1	S/. 100.00	S/. 100.00	S/. 18.00	S/. 118.00
7	Cables de luz	1	S/. 5.00	S/. 5.00	S/. 0.90	S/. 5.90
8	socate	1	S/. 15.00	S/. 15.00	S/. 2.70	S/. 17.70
9	Foco	1	S/. 16.00	S/. 16.00	S/. 2.88	S/. 18.88
						S/. 313.88

Fuente: Elaboración propia

**Tabla 3. 17 Materiales para la elaboración de la maqueta.**

N°	Materiales de trabajo	cantidad	Precio unitario	subtotal	IGV	Total
1	Ploteo	1	S/. 3.00	S/. 3.00	S/. 0.54	S/. 3.54
2	Carton maqueta	1	S/. 8.50	S/. 8.50	S/. 1.53	S/. 10.03
3	Carton corrugado	1	S/. 2.50	S/. 2.00	S/. 0.36	S/. 2.36
4	Silicona	2	S/. 5.00	S/. 40.00	S/. 7.20	S/. 47.20
5	Texturas	2	S/. 4.00	S/. 8.00	S/. 1.44	S/. 9.44
6	Cartulina	3	S/. 2.50	S/. 7.50	S/. 1.35	S/. 8.85
7	Microporoso verde texturado	2	S/. 3.00	S/. 6.00	S/. 1.08	S/. 7.08
8	Vegetacion	1	S/. 2.00	S/. 2.00	S/. 0.36	S/. 2.36
9	Madera cedro	3	S/. 1.30	S/. 3.90	S/. 0.70	S/. 4.60
10	Fuente de 5 voltios	1	S/. 40.00	S/. 40.00	S/. 7.20	S/. 47.20
11	Micas transparentes	3	S/. 1.50	S/. 4.50	S/. 0.81	S/. 5.31
12	Madera balsa	3	4	S/. 12.00	S/. 2.16	S/. 14.16
13	Cartulina duplex	3	S/. 2.50	S/. 7.50	S/. 1.35	S/. 8.85
						S/. 170.98

*Fuente: Elaboración propia*

### 3.7. PRUEBAS DE UNIDAD Y PRUEBAS DEL SISTEMA.

En este momento del ciclo de desarrollo nos encontramos en la necesidad de verificar el funcionamiento de la aplicación móvil y comprobar si cumple satisfactoriamente con los requerimientos previamente establecidos.

Las pruebas tienen como objetivo principal encontrar errores y fallas que se pueden haber creado en el momento del desarrollo e implementación de la aplicación. Los errores y fallas detectadas deberán ser depurados. Se presenta las distintas pruebas de unidad que se realizaran a la aplicación.

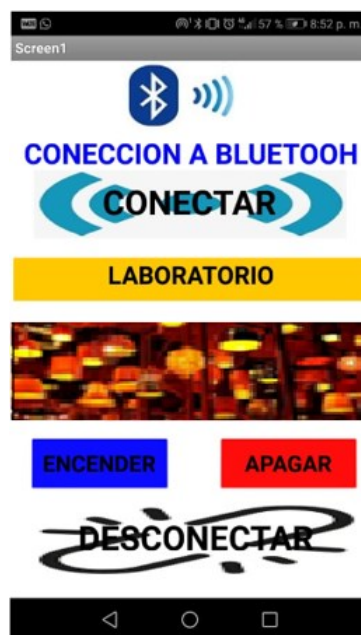
A continuación, se presenta las pruebas detalladas:

### 3.7.1. Establecer conexión Bluetooth

**Tabla 3. 18 Prueba unitaria: Establecer conexión Bluetooth.**

Prueba de unidad: Establecer conexión Bluetooth	
<b>PROYECTO</b>	PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL.
<b>FECHA DE REALIZACIÓN</b>	27/06/2019
<b>RESPONSABLE</b>	Gleydi Gallardo Vásquez
<b>OBJETIVO</b>	Comprobar que la conexión Bluetooth se establezca correctamente con el Arduino Uno (sistema de componentes electrónicos).
<b>DATOS DE INGRESO</b>	
<b>PROCEDIMIENTO DE PRUEBA</b>	
<ul style="list-style-type: none"> <li>• Ingresar a la aplicación.</li> <li>• Clic en el botón “Conectar”.</li> <li>• Selecciona la conexión Bluetooth del Arduino Uno.</li> </ul>	
<b>RESULTADO</b>	La aplicación establece la conexión Bluetooth satisfactoriamente y muestra la pantalla principal.

*Fuente: Elaboración propia*



**Figura 3. 22 Prueba unitaria: Establecer conexión Bluetooth**

*Fuente: Elaboración propia*

### 3.7.2. Verificar detección de movimiento

**Tabla 3. 19 Prueba unitaria: Verificar detección de movimiento.**

<b>Prueba de unidad: Verificar detección de movimiento.</b>	
<b>PROYECTO</b>	PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL.
<b>FECHA DE REALIZACIÓN</b>	28/06/2019
<b>RESPONSABLE</b>	Juliana Chinguel Zurita
<b>OBJETIVO</b>	Comprobar que la aplicación móvil notifique la detección de movimiento al usuario con mensaje, sonido y muestre en la interfaz principal un botón de color verde “HAY MOVIMIENTO” o en su defecto un botón de color rojo “NO HAY MOVIMIENTO”.
<b>DATOS DE INGRESO</b>	
<b>PROCEDIMIENTO DE PRUEBA</b>	
<ul style="list-style-type: none"> <li>• Ingresar a la aplicación móvil.</li> <li>• Muestra en la pantalla principal la detección del movimiento y envió de notificaciones.</li> </ul>	
<b>RESULTADO</b>	En la pantalla principal de la aplicación móvil se muestra un botón de color verde “HAY MOVIMIENTO” o en su defecto un botón de color rojo “NO HAY MOVIMIENTO”, y envía notificaciones al usuario emitiendo sonido y mensaje en la parte superior de la barra del dispositivo móvil.

*Fuente: Elaboración propia*



**Figura 3. 23 Prueba unitaria: Verificar detección de movimiento.**  
*Fuente: Elaboración propia*

### 3.7.3. Verificar encendido de luminaria

**Tabla 3. 20 Prueba unitaria: Verificar encendido de luminaria.**

<b>Prueba de unidad: Verificar encendido de luminaria.</b>	
<b>PROYECTO</b>	PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL.
<b>FECHA DE REALIZACIÓN</b>	29/06/2019
<b>RESPONSABLE</b>	Gleydi Gallardo Vásquez.
<b>OBJETIVO</b>	Comprobar que la aplicación encienda luminaria del centro de cómputo.
<b>DATOS DE INGRESO</b>	
<b>PROCEDIMIENTO DE PRUEBA</b>	
<ul style="list-style-type: none"> <li>• Clic en botón “Encender”.</li> <li>• La aplicación envía la orden de encender la luminaria del centro de cómputo, mediante la comunicación Bluetooth establecida con el Arduino Uno.</li> <li>• La luminaria del centro de cómputo debe encender.</li> </ul>	
<b>RESULTADO</b>	La aplicación enciende la luminaria del centro de cómputo.

*Fuente: Elaboración propia.*

### 3.7.4. Verificar apagado de luminaria

**Tabla 3. 21 Prueba unitaria: Verificar apagado de luminaria.**

Prueba de unidad: Verificar apagado de luminaria.	
PROYECTO	PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL.
FECHA DE REALIZACIÓN	29/06/2019
RESPONSABLE	Yessenia Marchena Estrada.
OBJETIVO	Comprobar que la aplicación apague luminaria del centro de cómputo.
DATOS DE INGRESO	
PROCEDIMIENTO DE PRUEBA	
<ul style="list-style-type: none"><li>• Clic en botón “Apagar”.</li><li>• La aplicación debe enviar la orden de apagar la luminaria del centro de cómputo, mediante la comunicación Bluetooth establecida con el Arduino Uno.</li><li>• La luminaria del centro de cómputo debe apagarse.</li></ul>	
RESULTADO	La aplicación apaga la luminaria del centro de cómputo.

*Fuente: Elaboración propia.*

### 3.7.5. Desconectar conexión Bluetooth

**Tabla 3. 22 Prueba unitaria: Desconectar conexión Bluetooth.**

Prueba de unidad: Desconectar conexión Bluetooth	
PROYECTO	PROTOTIPO DE UN SISTEMA DE CONTROL DE ILUMINACIÓN PARA CENTROS DE CÓMPUTO A TRAVÉS DE UN DISPOSITIVO MÓVIL.
FECHA DE REALIZACIÓN	29/06/2019
RESPONSABLE	Teresa Silva Otero.
OBJETIVO	Comprobar que la aplicación desconecta la conexión Bluetooth.
DATOS DE INGRESO	
PROCEDIMIENTO DE PRUEBA	
<ul style="list-style-type: none"><li>• Clic en botón “Desconectar”.</li><li>• Se desconecta la conexión Bluetooth establecida.</li></ul>	
RESULTADO	La aplicación desconecta la conexión Bluetooth con el Arduino Uno.

*Fuente: Elaboración propia*

## **CONCLUSIONES**

Con el desarrollo del prototipo se logrará el objetivo principal de este trabajo el mismo que permite controlar la energía eléctrica de un centro de cómputo y evitar el consumo innecesario de esta.

Se comprobó que el código para la conexión del circuito eléctrico y electrónico para los dos tipos de administración (Manual y Automático) funciona satisfactoriamente, tomando en consideración la iluminación requerida cuando el sensor PIR detecte la presencia de un cuerpo en movimiento.

La conectividad entre la aplicación móvil y el Arduino fue satisfactoria, esto gracias a que se cuenta con herramientas de hardware, software y código abierto, lo cual permitirá futuras integraciones y modificaciones en la estructura, obteniendo nuevos servicios y facilidades a los usuarios.



## **RECOMENDACIONES**

Se recomienda utilizar un dispositivo móvil smartphone con sistema operativo Android, para disfrutar de todas las funcionalidades y profundizar en la intensidad de la luz.

Se recomienda que el dispositivo móvil tenga una tarjeta con suficiente memoria para poder guardar el historial del consumo de energía eléctrica para realizar posteriores análisis comparativos y determinar el ahorro de energía.

## BIBLIOGRAFÍA

- Embedded Laboratory. (2018). *Getting Started with MQTT using Mosquitto*. Obtenido de Embedded Laboratory: <http://embeddedlaboratory.blogspot.com/2018/01/getting-started-with-mqtt-using.html>
- García Macías, J., & Marcillo Zambrano, C. (2017). Sistema Domótico Mediante Smartphone de la Iluminación en el Auditorio de la Carrera De Computación – Espam-MFL. *ESPAM-MFL*.
- Guerrero, J. (21 de setiembre de 2014). *Arduino Uno: Especificaciones y características*. Obtenido de PlusElectric: <https://pluselectric.wordpress.com/2014/09/21/arduino-uno-especificaciones-y-caracteristicas/>
- Huamán, J. (abril de 2017). *Control inteligente de sistemas e iluminación en edificios (Tesis de Máster en Ingeniería Mecánico-Eléctrica con Mención en Automática y Optimización)*. Universidad de Piura. Facultad de Ingeniería. Piura, Perú. Obtenido de <https://pirhua.udep.edu.pe/handle/11042/2897>
- López, C., Espinoza, M., & Barrientos Padilla, A. (2015). Implementación de una solución de domótica basado en las mejores soluciones y prácticas del mercado actual. *Universidad Peruana de Ciencias Aplicadas*.
- Master, S. (s.f.). *Definición de Aplicación*. Obtenido de <https://sistemas.com/aplicacion.php>
- Moises, B. A. (2018). *Internet de las cosas*. Obtenido de [https://www.editorialreus.es/static/pdf/primeraspaginas\\_9788429020380\\_internetdelas cosas.pdf](https://www.editorialreus.es/static/pdf/primeraspaginas_9788429020380_internetdelas cosas.pdf)
- Naylamp Mechatronics. (a). *Arduino Uno*. Obtenido de Naylamp Mechatronics: <https://naylampmechatronics.com/arduino-tarjetas/8-arduino-uno-r3-00008.html>
- Naylamp Mechatronics. (b). *Módulo PIR HC-SR501*. Obtenido de Naylamp Mechatronics: <https://naylampmechatronics.com/sensores-proximidad/55-modulo-de-deteccion-pir-hc-sr501.html>
- Naylamp Mechatronics. (c). *Módulo Bluetooth HC06*. Obtenido de Naylamp Mechatronics: <https://naylampmechatronics.com/inalambrico/24-modulo-bluetooth-hc06.html>
- Naylamp Mechatronics. (d). *Cable Dupont hembra a macho 20cm / 20Und*. Obtenido de Naylamp Mechatronics: <https://naylampmechatronics.com/accesorios-y-prototipado/81-cable-dupont-hembra-a-macho-20cm-x-20und.html>
- Naylamp Mechatronics. (e). *Protoboard*. Obtenido de Naylamp Mechatronics: <https://naylampmechatronics.com/accesorios-y-prototipado/14-protoboard-830.html>
- Naylamp Mechatronics. (f). *Módulo Relay*. Obtenido de Naylamp Mechatronics: <https://naylampmechatronics.com/drivers/31-modulo-relay-2-canales-5vdc.html>

Pedro, Á. P. (s.f.). *El internet de las cosas y las luces inteligentes*. Obtenido de <https://revistafeel.com.mx/feel-lifestyle/el-internet-de-las-cosas-y-las-luces-inteligentes/#>

Sergio, L. M. (2002). *Programación de aplicaciones web: Historia, principios básicos y clientes web*. España: Club Universitario.

Universidad de Salamanca. (s.f.). BÁSICOS APP INVENTOR: Manual de Introducción a AppInventor. Obtenido de <http://diarium.usal.es/igallego/files/2015/06/Basicos-APPIinventor-Manual-de-Introduccion.pdf>

## Anexo N° 1

### Estructura y Codificación del software

#### Glosario de términos – Robert.

#### Manual de usuario.

#### Codificación en Arduino

```
#include<SoftwareSerial.h>

unsigned long starMov = 0;
//unsigned long starNoMov = 0;
bool InicioDisparo = false;

const int PIR = 7;
int estadoPir = LOW;
int valor = 0;
int rele = 9;

void setup() {p
    pinMode(PIR, INPUT);
    pinMode(rele, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    if (Serial.available() > 0) {
        char datoRecibido = Serial.read();
        if (datoRecibido == 'A' || datoRecibido == 'a') {
            digitalWrite(rele, HIGH);
        }
        if (datoRecibido == 'B' || datoRecibido == 'b') {
            digitalWrite(rele, LOW);
        }
    }
    valor = digitalRead(PIR);
    if (valor == HIGH) {
        Serial.println(valor);
        estadoPir = HIGH;
        InicioDisparo = true;
        starMov = millis();
    }
    if (InicioDisparo && (millis() - starMov > 20000)) {
        digitalWrite(rele, LOW);
        InicioDisparo = false;
    }
    else {
        if (estadoPir == HIGH) {
            Serial.println(valor);
            estadoPir = LOW;
        }
    }
}
```